CLAIMS

What is claimed is:

1. A system for optimizing Network Interface Card (NIC) based data communications between an application and a destination through a NIC wherein said NIC and said application are on a host computer system and said destination is reachable through a port on said NIC, said system, comprising:

a token table associated with and for substantially each of said NICs on said host computer system, for communicating descriptors and control-path commands to the NIC,

a Notification Request Area (NRA) in a memory on said NIC for indicating when the NIC should notify host software that a said descriptor completes,

a Master Completion Queue in said memory on said host computer, having a list of completed items, said items on said list being used to store completion status of descriptors posted to work queues which are associated with a completion queue.

2. The system of claim 1 further comprising a memory deregistration system having;

a Memory Deregistration List (MDL) in-NIC memory containing memory handles to be deregistered,

an MDL Insert Kernel Agent counter in said host computer system, which indicates where in the MDL a next deregistered memory handle should be inserted,

a Memory Region Table (MRT) which is a host-resident table maintained by the Kernel Agent, containing all registration information for all memory regions registered on the NIC, and

a shadow MRT which is an in-NIC copy of active MRT entries for said NIC.

3. A method of optimizing data communications between an application and a source through a NIC, wherein said method comprises two processes, message descriptor posting and message descriptor processing.

4. The method of claim 3 operating on a system for optimizing Network Interface Connection (NIC) based data communications between an application and a destination

through a NIC wherein said NIC and said application are on a host computer system and said destination is reachable through a port on said NIC, said system, comprising:

a token table associated with and for substantially each of said NICs on said host computer system, for communicating descriptors and control-path commands to the NIC,

a Notification Request Area (NRA) in a memory on said NIC for notifying ___ when a said descriptor completes,

a Master Completion Queue in said memory on said host computer, having a list of completed items, said items on said list being requests for transferring data across said NIC,

memory deregistration system comprising a Memory Deregistration List (MDL) in-NIC memory containing memory handles to be deregistered, andMDL Insert Kernel Agent counter in said host computer system, which indicates where in the MDL a next deregistered memory handle should be inserted, a Memory Region Table (MRT) which is a host-resident table maintained by the Kernel Agent, containing all registration information for all memory regions registered on the NIC, and a shadow MRT which is an in-NIC copy of active MRT entries for said NIC.

5. The method of claim 3 wherein said message descriptor posting comprises:

receiving a call from a user application a VIA Provider Library (VIPL) function it wants while supplying said VIPL with a descriptor ,

linking the descriptor onto the appropriate Work Queue,

atomically incrementing a Token Table's Last_Inserted_Entry, and storing a representation of the result in a Local_Insert_Index of said Token Table,

using the descriptor to fill in a Post_Array fields of said Token Table,

determining if this is the first unaccepted Post_Array entry and if so, writing a new-entry-posted notification to an Out Post FIFO on an I/O bridge linking the NIC and the host computer system.

6. The method of claim 5 wherein prior to linking said descriptor onto said appropriate work queue, validating said descriptor's contents.

7. The method of claim 5 wherein prior to using the descriptor to fill in a Post_Array fields of said Token Table, determining if the Token Table is full, and if so, causing the system to wait until there is room on the token table before processing the application request.

8. The method of claim 3 wherein said message descriptor processing is initiated by an interrupt to a Message Unit from said Out Post FIFO on said NIC.

9. The method of claim 3 wherein the message descriptor processing comprises:
operating a primary Do loop of the Msg Unit which
Copies the_Token Table_Entries from Token Table.Post_Array;
and for each copied entry that is new, determining
whether this new entry contain a descriptor, and if so
fetching memory registration information for the memory region containing the descriptor and storing that information in a shadow copy of a memory registration table on the NIC;
Determining if this entry contains an indirect descriptor reference and if so,
copying the descriptor from application program associated memory;
building an I/O Resource Block (IORB) that represents this descriptor;
and calling a Vito Protocol component and passing said IORB to said Vito Protocol component;
if the entry does not contain an indirect descriptor reference, handling the entry as a control command, else
incrementing a counter for the last accepted entry and
Continue the Do loop until the last copied entry is already handled.

10. The method of claim 9 further comprising,

adjusting the number of token table entries copied per copy operation based upon load.

11. The method of claim 9 further comprising:
adjusting frequency of polling of the Token Table by the Message Unit based on measured load as determined by number of valid Post Array entries.

12. A method of memory registration messaging for operation on said system of claim 1 comprising:

receiving a call request to register an area of memory,

transferring data regarding said requested memory registration regarding its size to the Virtual Interface Provider Library,

calling a Kernel Agent with the data regarding said requested memory

allocating said memory to be registered via an entry in a memory registration table,

calling an Operating System to load all of th e memory space in said requested area of memory and to reserve said requested area in said Operating System's physical address reservation system

storing information related to said Operating System physical address reservation system.

13. The method of claim 12 wherein before said storing step, determining if the memory space spans less than or equal to the maximum area for local data and if so directly storing the Operating System's physical address reservation system information, but if not, storing a reference to the Operating System's physical address system.

14. The method of claim 12 wherein said information related to said Operating System physical address reservation system is returned to the application as a memory handle.

15. The method of claim 12 wherein an index of a memory registration table is returned to the Virtual Interface Provider Library.

16. A method of Memory deregistration for use in the system of claim 1 by which memory owned by a process is let go and returned to the system for other processes to use comprising:

not handling memory deregistration until idle time or validation of a previously registered region.

17. A system as set forth in claim 1 wherein there exists a master completion queue for each NIC and each said master completion queue can reference any completion queue associated with the NIC.

18. A system as set forth in claim 2 wherein there is a send and a receive work queue and a completion queue for each application program and wherein a notification request area exists for each completion queue and for said master completion queue, said notification request area providing notice to requestors of completions.

19. A method of Memory deregistration for use in the system of claim 2 by which memory owned by a process is let go and returned to the system for other processes to use comprising:

not handling memory deregistration until idle time or validation of a previously registered region.

20 A system as set forth in claim 2 wherein there exists a master completion queue foe each NIC and each said master completion queue can reference any completion queue within the NIC.

21. A system as set forth in claim 2 wherein there is a send and a receive work queue and a completion queue for each application program and wherein a notification request area exists for each completion queue and for said master completion queue, said notification request area providing notice to requestors of completions.